

Diagrama entidad relación

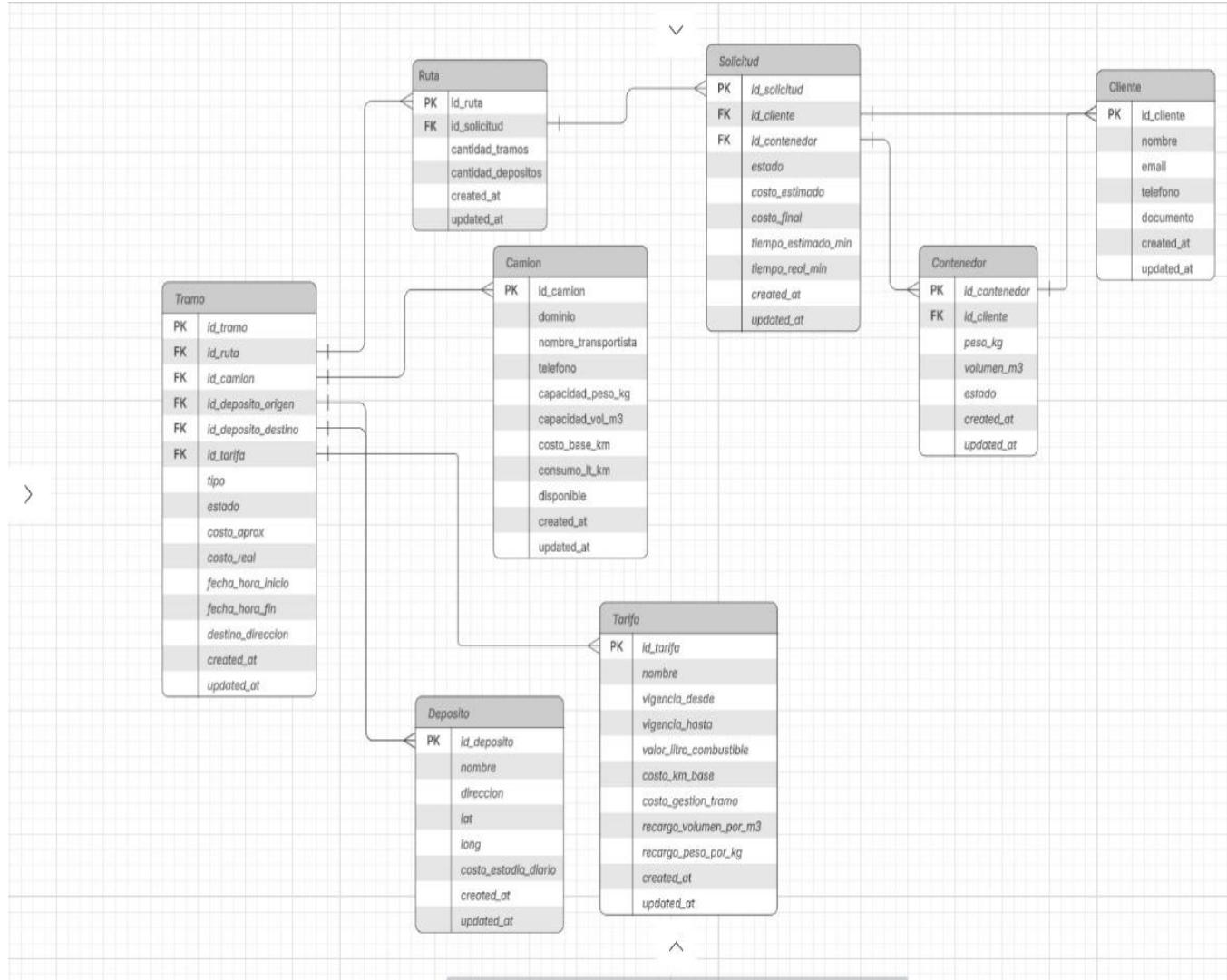
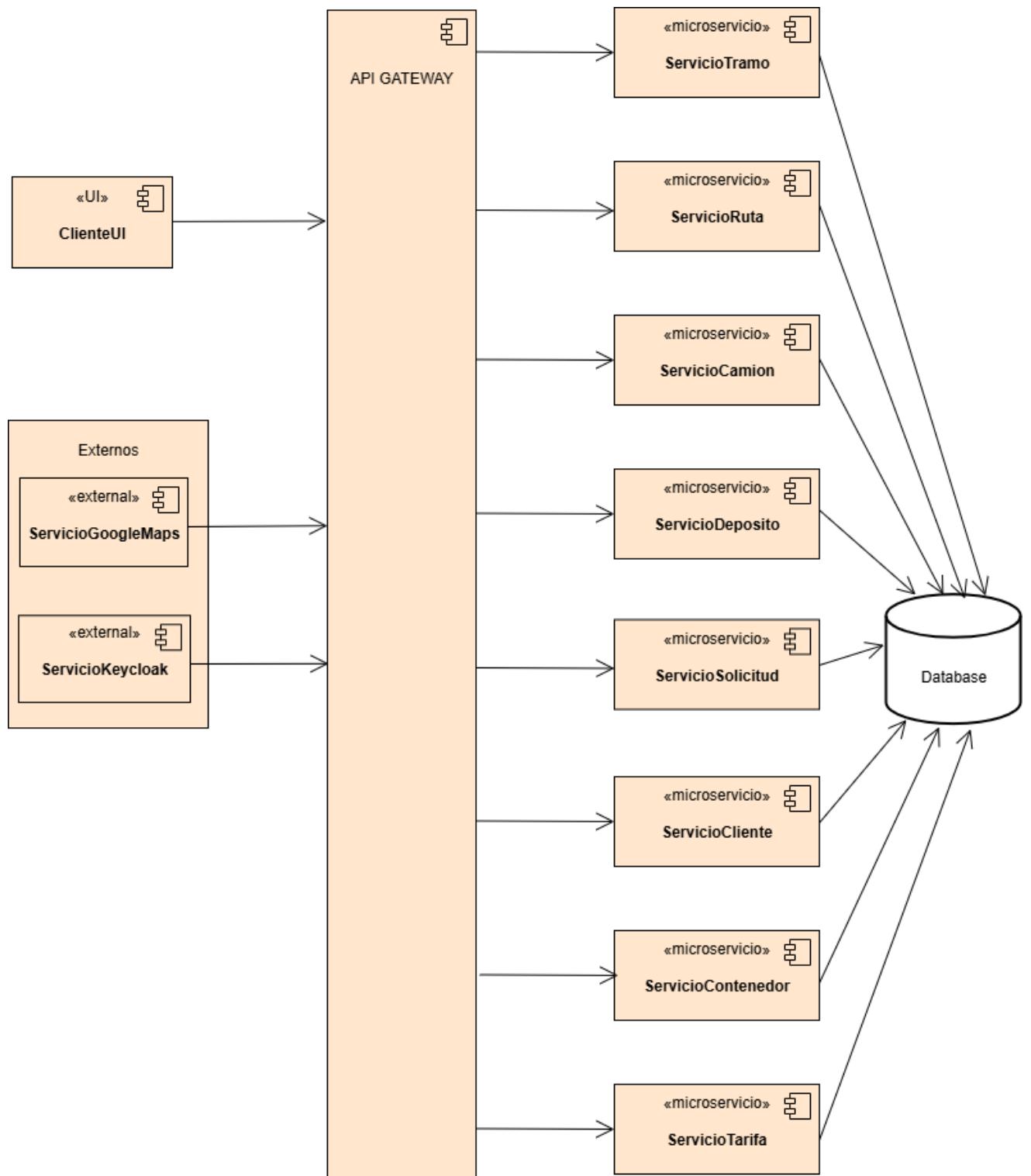


Diagrama de microservicios



## Diseño de Microservicios Funcionales

### Arquitectura general

El sistema se compone de microservicios independientes desplegados en contenedores Docker, comunicados entre sí mediante HTTP REST y autenticados vía Keycloak con tokens JWT. Los servicios se exponen a través de un API Gateway que valida las credenciales y enruta las solicitudes según el rol del usuario.

Microservicios principales:

- - Cliente Service
- - Contenedor Service
- - Camión Service
- - Solicitud y Ruta Service
- - Depósito Service
- - Tarifa / Configuración Service
- - Auth Service (Keycloak)

### 1. Cliente Service

Recurso: /clientes

Descripción: Gestiona la información de los clientes y sus solicitudes de traslado.

Endpoint	Método	Rol	Descripción	Datos Entrada	Datos Salida
/clientes	POST	Cliente / Operador	Registra un nuevo cliente en el sistema	Nombre, email, teléfono, dirección	ID cliente, fecha alta
/clientes/{id}	GET	Operador	Consulta los datos de un cliente específico	—	Datos completos del cliente
/clientes/{id}/solicitudes	GET	Cliente / Operador	Obtiene las solicitudes asociadas al cliente	—	Listado de solicitudes con estado

### 2. Contenedor Service

Recurso: /contenedores

Descripción: Administra los contenedores, su peso, volumen, cliente asociado y estado actual.

Endpoint	Método	Rol	Descripción	Datos	Datos Salida
----------	--------	-----	-------------	-------	--------------

				<b>Entrada</b>	
/contenedores	POST	Operador	Crea un nuevo contenedor	Peso, volumen, clientID	ID contenedor, estado inicial
/contenedores/{id}	GET	Cliente / Operador	Consulta el estado actual del contenedor	—	Estado (en depósito, en tránsito, entregado)
/contenedores	GET	Operador	Lista contenedores con filtros	Query params	Listado de contenedores

### 3. Camión Service

Recurso: /camiones

Descripción: Administra la flota de camiones, su disponibilidad, capacidad y transportistas asociados.

<b>Endpoint</b>	<b>Método</b>	<b>Rol</b>	<b>Descripción</b>	<b>Datos Entrada</b>	<b>Datos Salida</b>
/camiones	POST	Operador	Registra un camión nuevo	Dominio, capacidad (peso y volumen), consumo, transportista	ID camión
/camiones/disponibles	GET	Operador	Obtiene camiones no asignados	—	Lista de camiones disponibles
/camiones/{id}/estado	PATCH	Transportista	Cambia el estado (en viaje o disponible)	Estado nuevo	Camión actualizado

### 4. Solicitud y Ruta Service

Recurso: /solicitudes, /rutas, /tramos

Descripción: Gestiona las solicitudes de traslado, la generación de rutas y la asignación de tramos y camiones.

<b>Endpoint</b>	<b>Método</b>	<b>Rol</b>	<b>Descripción</b>	<b>Datos Entrada</b>	<b>Datos Salida</b>
/solicitudes	POST	Cliente	Crea una nueva solicitud de traslado	Origen, destino, datos contenedor/cliente	ID solicitud, costo y tiempo estimado
/solicitudes/{id}	GET	Cliente /	Consulta	—	Datos

		Operador	detalle y estado actual de la solicitud		completos de la solicitud
/solicitudes	GET	Operador	Lista solicitudes por estado o cliente	Query params	Listado con estado y contenedor
/solicitudes/{id}/ruta	POST	Operador	Asigna una ruta con sus tramos	Lista de puntos (origen, depósitos, destino)	Rutas y tramos generados
/tramos/{id}/camion	PATCH	Operador	Asigna camión a tramo	ID camión	Tramo actualizado
/tramos/{id}/inicio	PATCH	Transportista	Marca inicio de tramo	FechaHoraInicio	Estado: iniciado
/tramos/{id}/fin	PATCH	Transportista	Marca fin de tramo	FechaHoraFin	Estado: finalizado, costo real

Integración externa: Este servicio consume la API de Google Maps Directions para calcular distancias y tiempos entre coordenadas, y consulta los servicios de Tarifa y Camión para determinar el costo total.

## 5. Depósito Service

Recurso: /depositos

Descripción: Administra los depósitos intermedios donde se almacenan temporalmente los contenedores.

Endpoint	Método	Rol	Descripción	Datos Entrada	Datos Salida
/depositos	POST	Operador	Registra un nuevo depósito	Nombre, dirección, coordenadas, costo diario	ID depósito
/depositos	GET	Operador	Lista todos los depósitos	—	Listado de depósitos con ubicación
/depositos/{id}/contenedores	GET	Operador	Consulta contenedores almacenados	—	Listado de contenedores y fechas

## 6. Tarifa / Configuración Service

Recurso: /tarifas

Descripción: Administra las tarifas, parámetros de costo por kilómetro, combustible y estadía.

Endpoint	Método	Rol	Descripción	Datos Entrada	Datos Salida
/tarifas	GET	Operador	Obtiene tarifas vigentes	—	Valores actuales (por km, litro, estadía)
/tarifas	POST	Operador	Crea o actualiza tarifas	Tipo, valor, fecha vigencia	Tarifa registrada
/parametros	GET	Operador	Consulta parámetros globales	—	Configuración actual del sistema

## 7. Auth Service (Keycloak)

Descripción: Servicio encargado de la autenticación y autorización de usuarios. No posee endpoints propios desarrollados en el backend, ya que se integra directamente con el flujo de autenticación de Keycloak.

Operación	Descripción
Login / Token JWT	Validación de credenciales y emisión del token de acceso
Validación de rol	Control de acceso a los endpoints de cada microservicio

## Resumen de Roles y Permisos

Servicio	Cliente	Operador / Administrador	Transportista
Cliente	Crear / Ver propio	CRUD	—
Contenedor	Ver estado	CRUD	—
Camión	—	CRUD / Asignar	Cambiar estado
Solicitud / Ruta	Crear / Ver propia	Ver todas / Asignar	Iniciar / Finalizar tramo
Depósito	—	CRUD / Consultar contenedores	—
Tarifa	—	CRUD	—

## Datos compartidos

Los microservicios se comunican mediante peticiones REST internas autenticadas por token y comparten identificadores de entidades clave (cliente, contenedor, solicitud, tramo). Cada microservicio mantiene su propia base de datos independiente, garantizando bajo acoplamiento, escalabilidad y una arquitectura alineada a las buenas prácticas del desarrollo backend basado en microservicios.